Bug 3395
Should for-of throw if iterator.return does not return `{done: true}`?

- [https://bugs.ecmascript.org/show_bug.cgi?id=3395](https://bugs.ecmascript.org/show_bug.cgi?id=3395)
- Dave’s original presentation:
  - On abrupt exit, for...of looks for return method.
  - If present, it calls the method with no arguments.
  - If the result has falsy done property, throw an error.
Current spec

• Throws if return result is an object, but doesn’t check for ‘done” property

• Why not:
  – No where else in the spec. do we do such a structural validity check on IteratorResult objects.
  – It’s be useful to have iterators that refuse to be closed
let arrowOfTime = function*() {
  let now = 0;
  while (true) yield now++;
} ();
arrowsOfTime.return = () => {done:false};
function eon1 (aOfT) {
    for (t of aOfT) {
        dayInTheLife1(t);
        if (endOfEon) break;
    }
}

function eon2 (aOfT) {
    for (t of aOfT) {
        dayInTheLife2 (t);
        if (endOfEon) break;
    }
}
function timeMarchesOn(aOfT){
    eon1(aOfT);
    eon2(aOfT);
    ...
}

timeMarchesOn(arrowOfTime);